Electrocap: A Shape Inverse Model for an Electro-Capillarity Process

Jérôme Monnier, Patrick Chow-Wing-Bom

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Electrocap: A Shape Inverse Model for an Electro-Capillarity Process

Jérôme Monnier — Patrick Chow-Wing-Bom

N° 5617

Juillet 2005

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--5617--FR+ENG

Electrocap: A Shape Inverse Model for an Electro-Capillarity Process

Jérôme Monnier *, Patrick Chow-Wing-Bom †

Thème NUM — Systèmes numériques
Projet Idopt

Rapport de recherche n° 5617 — Juillet 2005 — 30 pages

Abstract: Electrocap is a freeware modelling an electrified droplet at equilibrium and wetting a solid surface. The model consists to seek the drop shape by minimizing its total energy (capillary, electrostatic and gravitational). To this end, the model is based on the classical shape optimal design method. In this paper, we derive the equations and the shape gradient; we detail the shape optimization algorithm implemented and present some numerical results. The code is based on the public mesh generator Bamg and the public C++ finite element library Rheolef.

Key-words: Shape optimal design, C++ library, Electro-Wetting, Energy minimization

* Jerome.Monnier@imag.fr
† chow@imag.fr

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Electrocap: Un modèle inverse de forme destiné à l'electro-capillarité

Résumé : Electrocap est un logiciel libre d'optimisation de forme modélisant une goutellette electrifiée à l'équilibre et mouillant une surface plane solide. Le modèle consiste à chercher une forme de goutte minimisant son énergie totale (capillaire, electro-statique et gravitationnelle). Dans ce papier, nous présentons les équations, le gradient de forme; nous détaillons le processus d'optimisation implémenté et nous présentons quelques résultats numériques. Le code est basé sur le mailleur Bamg et la bibliothéque éléménts finis C++ Rheolef.

Mots-clés : Optimisation de forme, librairie C++, Electro-mouillage, minimisation d'énergie

1 Introduction

Electro-wetting can be defined as a tool for spreading liquid droplet (e.g. water) on hydrophobic solid surfaces (e.g. polymer film). This quite recent technique, [1], presents very attractive properties for manipulation of tiny liquid volumes, as it is done in biotechnologies. The principle of electro-wetting is to apply an electric field between the conductor liquid droplet and the solid surface in order to change the droplet spreading on the surface. Given the liquid volume, the main feature to describe the droplet is the wetting angle.
Few articles treat of the experimental aspects of electro-wetting and present some analytical analysis, see e.g. [1], [13], [3] and references therein. A property of electro-wetting still badly understood by physicists is the contact angle saturation, [13]. This limiting phenomena is the following. When increasing the applied electric voltage, the liquid droplet spreads onto the solid and the wetting angle decreases. Nevertheless, this is true only until a critical value of the applied voltage. Up to this critical value, the plane capacitor approximation(Lippman's equation remains valid. For higher values, one observes a saturation of the wetting angle and for even higher values, instabilities of the contact line liquid-solid-gas appear. Few hypothesis have been made to explain the saturation phenomena. Let us cite air ionization near the edge, [13], and electrostatic edge effects near the edge [4]. All these explanations to the limiting phenomena are still under investigations and the full modelling of electro-wetting is still an open problem.

In the present study, we propose a shape optimization approach (see e.g. [5], [8]) in order to model this steady-state free surface flow (the electrified drop shape). We assume that all shapes are 2D axisymmetric. We seek the drop shape such that it minimizes its total energy. The total energy is the sum of the capillary energy, the gravitational energy and the electrostatic energy. The mathematical and numerical formalisms presented in this article remain valid for 3D shapes. Of course, in 3D case the implementation is much more complex and time-consuming than the present 2D axisymmetric case.
ElectroCap is a freeware written in C++. It uses the public C++ finite element library Rheolef and the public mesh generator Bamg.

In a microfluidics point of view, the numerical results we obtain are consistent with the plane capacitor approximation (Lippmann's equation). This means that up to a critical voltage, we retrieve the experimental results; and for higher voltages, we do not retrieve the wetting angle saturation but a total spreading of the drop, like Lippmamn's equation predicts. No wetting angle saturation has been obtained with the present model.

The paper is organized as follows. In Section 2, we present the electro-wetting process considered and recall the plane capacitor approximation. In Section 3, we derive the mathematical model. It is a shape inverse problem: we seek the drop shape such that it minimizes its total energy. The energy depends on the electric field, which is solution of an external partial differential equation. The liquid volume being given and constant, we consider it as an equality constraint. Finally, the problem consists to find a min-max solution (saddle

point) of an augmented lagrangian. Numerically, the solution is computed using the Uzawa's algorithm and a Quasi-Newton optimization algorithm (BFGS). In Section 4, we define the mathematical framework of shape optimization and we derive the shape derivative of the augmented lagrangian (Theorem 1). To this end we introduce the classical adjoint state equation. In section 5, we detail the discretization of the problem. The partial differential equation is solved using a standard linear (P1) Lagrange finite element method. The shape parameters are defined, the shape deformation basis is defined, then the shape gradient can be expressed and the optimization parameters obtained. The full optimization process is presented in Section 6. The program structure is presented in Section 7. In Section 8, we present some numerical results validating the present approach and the code. Finally, we present in appendix a short user manual.

2 Electrowetting Process

We consider the electrowetting process presented in Fig. 1.

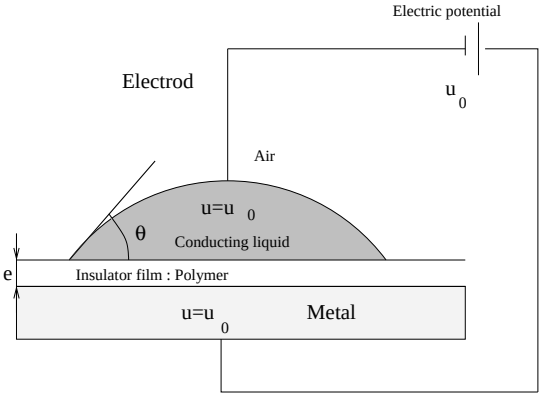

Figure 1: Electrowetting process

We denote by σ_{LS}, σ_{SG} and σ_{LG} the surface tension coefficients of the liquid-solid interface, solid-gas interface and liquid-gas interface respectively. We denote by θ the wetting angle.

When the applied electrical potential u_0 is null, the Young's equation gives:

$$cos(\theta_0) \;=\; \frac{\sigma_{SG} - \sigma_{LS}}{\sigma_{LG}}$$

where θ_0 is the wetting angle at $u_0 = 0$.
Under the assumption that the system behaves as a plane capacitor with boundary effects

negligible, the drop shape obeys to the Young equation with the surface tension coefficient modified as follows, [1]:

$$\sigma_{LS}(u_0) = \sigma_{LS} - \frac{\varepsilon_0\varepsilon_1}{2e}u_0^2$$

where e is the insulator thickness, ε_0 and ε_1 are the dielectric constant.
Also, we have, [1]:

$$cos(\theta) = cos(\theta_0) + \frac{\varepsilon_0\varepsilon_1}{2\sigma_{LG}\ e}u_0^2$$

This last equation is called Lippmann's equation too.

Let us notice that this law forecasts a total spreading when the potential increases. Nevertheless, if u_0 is greater a critical value u_{cr}, physicists observe a locking phenomena limiting the spreading of the droplet on the polymer film. Such experiments are studied in [1], [13], [3]. In other respect, let us point out that the application of the present experiment to variable focal lens is studied in [2].

The aim of the present study is to model and compute numerically the liquid drop shape for u_0 lower than such critical value u_{cr}. The main features are the wetting angle θ, and the curvature κ of the liquid surface, particularly near the contact line liquid-solid-gas.

3 Mathematical Modeling

We model the electro-wetting process described in previous section as a shape inverse problem.

Assumption 1 *i) The applied electrical potential u_0 is continuous.*
ii) The liquid drop is a perfect conductor.
iii) The drop geometry is 2D axisymmetric.
iv) Electrostatic effects are negligible far away from the drop.
v) For $u_0 = 0$, the liquid wets partially the polymer (the spreading coefficient is negative).

Notations. We denote by -see Figure 2-: $u(x)$ the electrical potential at point x, ω_0 the liquid drop, ω_1 the polymer domain, ω_2 the artificially bounded gas domain and γ_{ext} its external boundary. The external boundary γ_{ext} is supposed to be far enough from the liquid drop.
We denote by γ_{LS}, γ_{SG} and γ_{LG} the liquid-solid interface, solid-gas interface and liquid-gas interface respectively.
We set: $\omega = \omega_1 \cup \omega_2 \cup \gamma_{SG}$. We have: $\partial\omega_0 = \gamma_{Lz} \cup \gamma_{LG} \cup \gamma_{LS}$ and $\partial\omega = \gamma_0 \cup \gamma_{Sz} \cup \gamma_{LG} \cup \gamma_{Gz} \cup \gamma_{ext}$; with $\gamma_z = \gamma_{Gz} \cup \gamma_{Lz} \cup \gamma_{Sz}$.
We set: $B = \omega_0 \cup \omega \cup \gamma_{LG} \cup \gamma_{LS}$.
The liquid domain ω_0 will be variable; on the other hand, the domain B is given and fixed.

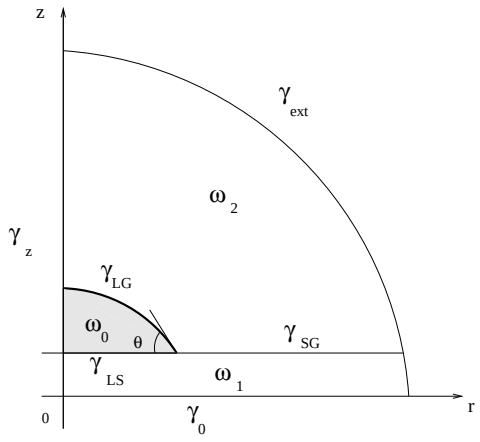

Figure 2: 2D axisymmetric droplet. Notations

The questions we answer numerically are the following. Given the electric potential u_0, what is the drop shape?, what is the wetting angle value θ?

Shape inverse formulation

We model this steady-state free surface problem as a shape inverse problem. We follow the approach done in [4].

The total energy $\mathcal{E}$ is the sum of the gravitational energy, the capillary energy and the electrostatic energy. In 3D case, its expression is, see e.g. [3]:

$$\mathcal{E}_{\omega_0} = \mathcal{E}^{grav}_{\omega_0} + \mathcal{E}^{cap}_{\omega_0} + \mathcal{E}^{elec}_{\omega_0}$$

with the gravitational energy,

$$\mathcal{E}^{grav} \;=\; \rho\, g \int_\omega z dx$$

with the capillary balance energy,

$$\mathcal{E}^{cap} \;=\; \int_{\gamma_{LS}} (\sigma_{LS} - \sigma_{GS}) ds \;+\; \int_{\gamma_{LG}} \sigma_{LG} ds$$

and the electrostatic energy,

$$\mathcal{E}^{elec} \;=\; -\,\frac{1}{2} \int_\omega \varepsilon\, |\nabla u|^2 dx$$

where: ρ is the liquid density, g is the gravity constant, $\varepsilon = \varepsilon_i$ in $\omega_i,\ i = 1, 2$, ε_i is the relative dielectric permittivity of ω_i i.e. $\varepsilon_0 \varepsilon_i,\ i = 1, 2$ is the polymer and the gas permittivity

respectively.

The shape inverse formulation is:

$$\left\{ \begin{array}{l} \text{Find } \omega_0^\star \text{ such that:} \\ \mathcal{E}_{\omega_0^\star} \ = \ \min\limits_{(\omega_0;\ \int_{\omega_0} dx = vol)} \mathcal{E}_{\omega_0} \end{array} \right.$$

where vol is the given drop volume.

We set $u_i = u|_{\omega_i},\ i = 1, 2$. Then, the potential u_i is solution of:

$$-\, div(\varepsilon_i \nabla u_i) \ = \ 0 \quad \text{in } \omega_i,\ i = 1, 2 \tag{1}$$

with the following Dirichlet boundary conditions:

$$\left\{ \begin{array}{rcll} u_1 & = & u_0 & \text{on } \gamma_{LG} \\ u_2 & = & u_0 & \text{on } \gamma_{LS} \\ u_2 & = & 0 & \text{on } \gamma_0 \end{array} \right. \tag{2}$$

On the solid-gas interface, we have the transmission boundary conditions:

$$\left\{ \begin{array}{rcll} u_1 & = & u_2 & \text{on } \gamma_{SG} \\ \varepsilon_1 \nabla u_1 n_1 & = & -\varepsilon_2 \nabla u_2 n_2 & \text{on } \gamma_{SG} \end{array} \right. \tag{3}$$

On the artificial boundary $\gamma_{ext} = \gamma_{ext}^1 \cup \gamma_{ext}^2$, we impose:

$$\varepsilon_i \nabla u_i n_i \ = \ 0 \qquad \text{on } \gamma_{ext}^i,\ i = 1, 2 \tag{4}$$

Therefore, the present mathematical problem is a shape optimal control problem of a system governed by a linear steady-state partial differential equation.

2D axisymmetric equations As mentioned previously, we assume that the drop shape is 2D axisymmetric. We present below the weak formulations of the model. We set:

$$X_0(\omega) = \{v \in H^1(\omega);\ v = 0 \text{ on } \gamma_0 \cup \gamma_{LS} \cup \gamma_{LG}\}$$

$$X_t(\omega) = \{v \in H^1(\omega);\ v = 0 \text{ on } \gamma_0;\ v = u_0 \text{ on } \gamma_{LS} \cup \gamma_{LG}\}$$

The weak formulation of (1)-(4) in the 2D axisymmetric case is:

$$\left\{ \begin{array}{l} \text{Find } u^\omega \in X_t(\omega) \text{ such that} \\ \forall v \in X_0(\omega),\ a_\omega(u^\omega, v) \ = \ 0 \end{array} \right. \tag{5}$$

where

$$a_\omega(u, v) \ = \ \int_\omega \varepsilon\ r\ \ < \nabla u, \nabla v > \ dx,$$

$x = (r, z)$ and $< ., . >$ is the inner product of $I\!R^2$.

In vertu of Lax-Milgram theorem, state equation (5) has one and only solution $u^\omega \in X_t(\omega)$.

The shape inverse problem. In its dimensionless form, the drop energy is:

$$\mathcal{E}_{\omega_0}(u^\omega) \;=\; \alpha \int_\omega z\; dx \;+\; \int_{\gamma_{LG}} r\; ds \;+\; \mu \int_{\gamma_{LS}} r\; dr \;-\; \delta \int_\omega \varepsilon\; r\; |\nabla u^\omega|^2\; dx \qquad (6)$$

where u^ω is the unique solution of (5), $\alpha = \frac{\rho g (L^*)^2}{\sigma_{LG}}$, $\mu = -cos(\theta_0) = \frac{\sigma_{LS} - \sigma_{GS}}{\sigma_{LG}}$, $\delta = \frac{1}{2\sigma_{LG} L^*}$ and L^* is a characteristic length (typically $L^* \approx 10^{-4} - 10^{-3}$ m).

We set the cost function by:

$$j(\omega) \;=\; \mathcal{E}_{\omega_0}(u^\omega) \qquad (7)$$

We denote by $\mathcal{D}$ the admissible domain space (the definition of $\mathcal{D}$ is detailed in next section). The shape optimal inverse problem is:

$$\left\{ \begin{array}{l} \text{Find } \omega^\star \in \mathcal{D} \text{ such that} \\ j(\omega^\star) \;=\; \min\limits_{(\omega;\; \int_{\omega_0} r dx = vol/2\pi)} j(\omega) \end{array} \right. \qquad (8)$$

Let us point out that the variable is not the whole domain ω but more precisely the liquid-gas interface γ_{LG}, see Fig. 2.

We assume that Problem (8) admits at least one solution.

The Augmented Lagrangian. Problem (8) is an optimization problem under an equality constraint. Thus, classically we introduce the augmented lagrangian $L_\tau : \mathcal{D} \times I\!R \longrightarrow I\!R$, defined by, see e.g. [7]:

$$L_\tau(\omega, \lambda) \;=\; j(\omega) + \lambda c(\omega) + \tau c(\omega)^2 \qquad (9)$$

where $c(\omega)$ is the volume constraint,

$$c(\omega) = \int_{\omega_0} r dx - \frac{vol}{2\pi} = \int_B r dx - \int_\omega r dx - \frac{vol}{2\pi}, \qquad (10)$$

λ is the Lagrange multiplier and τ is a penalty parameter.
Then, the shape optimal inverse problem (8) is formulated as the saddle-point problem:

$$\left\{ \begin{array}{l} \text{Find } (\omega^\star, \lambda^\star) \in \mathcal{D} \times I\!R \text{ such that} \\ L_\tau(\omega^\star, \lambda^\star) \;=\; \min\limits_\omega \max\limits_\lambda L_\tau(\omega, \lambda) \end{array} \right. \qquad (11)$$

We will solve (11) using Uzawa's algorithm. This algorithm uses a gradient type algorithm (BFGS) which requires to compute the shape derivative of the cost function, $\frac{dj}{d\omega}(\Omega)$, and the shape derivative of the constraint, $\frac{dc}{d\omega}(\Omega)$. The expressions of these shape derivatives are presented in next section.

4 Shape Derivatives

In this section, we define the admissible domain space $\mathcal{D}$ (Lipschitz domains), we use the classical definition of shape derivatives based on domain deformations (method of transport with $\mathcal{C}^1$ transformations). We refer to [10, 5]; definitions of [6, 9] are used.
We prove the differentiability of the cost function j and the constraint function c with respect to domain ω. Then, by introducing the adjoint state equation (in our case the adjoint state vanishes), we obtain the differential of j and c (Theorem 1). The shape derivative of the augmented lagrangian L_τ follows.
In next section, using a finite element discretization, these differential expressions leads to the shape gradients.

4.1 Mathematical framework: domain variations and shape derivatives

We consider family of Lipschitz domains. We define the space of admissible domains and the derivative with respect to the domain in a classical manner. The domain space is the set of domains homeomorphic to a reference domain. The transformations are $\mathcal{C}^1$ homeomorphisms, regularity required to well define the (volume and surface) transported integrals. The shape derivative of a real valued function is the derivative of the transported function with respect to the transformation. We refer to [10, 5]; we follow below the definitions and properties presented in [6, 9].

Admissible domain space. Let $\hat{\Omega}$, a bounded open subset of $I\!R^2$ with a Lipschitz boundary, be the reference domain: $\hat{\Omega} = \Omega_1 \cup \hat{\Omega}_2 \cup \hat{\Gamma}_{SG}$. Ω_1 represents the solid part and $\hat{\Omega}_2$ represents the gas part.
We distinguish the variable part of $\hat{\Omega}$ from its fixed part, see Fig. 3. We set: $\partial\hat{\Omega} = \hat{\Gamma}_{Var} \cup \Gamma_{Fix}$ where $\hat{\Gamma}_{Var} = \hat{\Gamma}_{LG} \cup \hat{\Gamma}_{LS}$ is the variable boundary and Γ_{Fix} is the fixed boundary. We denote by B_{int} a neighborhood of $\hat{\Gamma}_{Var}$, B_{int} large enough, see Fig. 3.

We set the function space:

$$\hat{\mathcal{F}} = \{\hat{F},\ \hat{F} \text{ bijection of } \hat{\Omega} \text{ onto } \hat{F}(\hat{\Omega});\ \hat{F} \in \mathcal{C}^1(\bar{\hat{\Omega}}, I\!R^d),\ \hat{F}^{-1} \in \mathcal{C}^1(\bar{\hat{F}}(\hat{\Omega}), I\!R^d)\} \qquad (12)$$

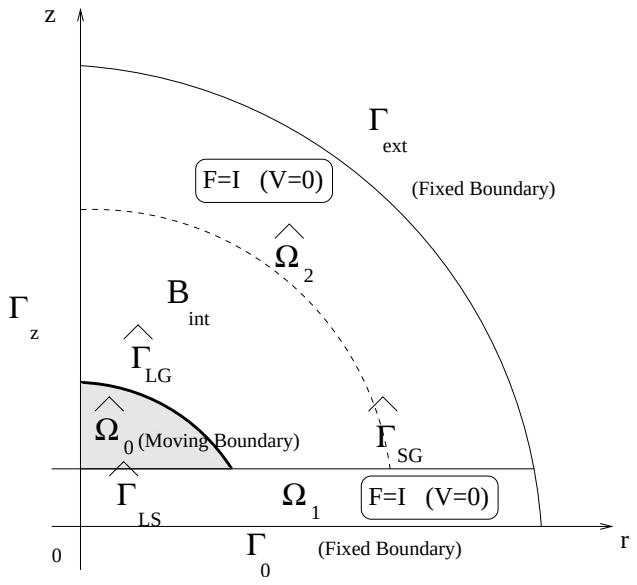

Figure 3: The reference domain $\hat{\Omega}$

and its affine subspace: $\hat{\mathcal{F}}_0 = \{\hat{F} \in \hat{\mathcal{F}};\ \hat{F} = I \text{ in } \hat{\Omega} \setminus B_{int}\}$, where I denotes the identity of $I\!R^d$.

Then, we define the admissible domains space $\mathcal{D}$ as follows

$$\mathcal{D} = \{\omega = \hat{F}_0(\hat{\Omega});\ \hat{F}_0 \in \hat{\mathcal{F}}_0\} \tag{13}$$

One knows that if $\hat{F}$ is close enough to I in $\hat{\mathcal{F}}_0$ ($(\hat{F} - I)$ small enough) then $\hat{F}(\hat{\Omega})$ is an open set of $I\!R^2$ with a Lipschitz boundary and $F(\hat{\Gamma}_{Var}) \subset B_{int}$.

Shape derivative of a real valued function For $\hat{F}_0 \in \hat{\mathcal{F}}_0$, $(\hat{F}_0 - I)$ small enough, we define the domain Ω by $\Omega = \hat{F}_0(\hat{\Omega})$ and $\Gamma_{Var} = \hat{F}_0(\hat{\Gamma}_{Var})$. We set the homeomorphisms space defined in Ω, Fig. 4: $\mathcal{F} = \{F,\ F = \hat{F} \circ \hat{F}_0^{-1},\ \hat{F} \in \hat{\mathcal{F}}\}$, and its affine subspace: $\mathcal{F}_0 = \{F,\ F = \hat{F} \circ \hat{F}_0^{-1},\ \hat{F} \in \hat{\mathcal{F}}_0\}$.

Let $F \in \mathcal{F}_0$, we define $\omega = F(\Omega)$ and $V \in \mathcal{C}^1(\bar{\Omega}, I\!R^d)$ by: $V = F - I$. We have $V = 0$ in $\hat{\Omega} \setminus B_{int}$.

For a given cost function j, $j : \omega \in \mathcal{D} \mapsto j(\omega) \in I\!R$, we define the "transported" cost function $\bar{j}$ by: $\bar{j} : \mathcal{F}_0 \to I\!R \ : \ F \mapsto \bar{j}(F) = j(F(\Omega)) = j(\omega)$. Then, the derivative with

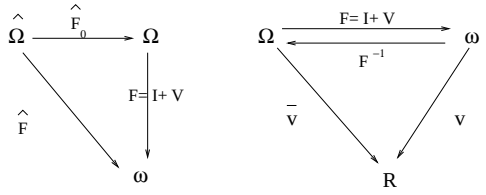

Figure 4: Change of variables

respect to the domain is defined as follows (see e.g. [10, 6] for more details):

$$\frac{dj}{d\omega}\ (\Omega)\cdot V \ = \ \frac{d\bar{j}}{dF}\ (I)\cdot V\ , \quad \forall V \in \mathcal{C}^1(\bar{\Omega}, I\!R^d) \tag{14}$$

4.2 Shape derivatives

We present below the expressions of the exact differentials with respect to the shape ω.

Theorem 1 *There exists $\mathcal{V}_I$, a neighborhood of I in $\mathcal{F}_0$, such that:*

i) the cost function $j : \mathcal{D} \longrightarrow I\!R;\ \omega \mapsto j(\omega) = \mathcal{E}_{\omega_0}(u^\omega)$ belongs to $\mathcal{C}^1$ for all $\omega = F(\Omega)$, $F \in \mathcal{V}_I$. And for all $V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$, we have:

$$\frac{dj}{d\omega}(\Omega).V \ = \ \frac{\partial \mathcal{E}_{\Omega_0}}{\partial \omega}(u^\Omega).V \tag{15}$$

with u^Ω the solution of the state equation (5) posed in Ω and

$$\begin{aligned}
\frac{\partial \mathcal{E}_{\Omega_0}}{\partial \omega}(u^\Omega).V \ = \ & \alpha \int_\Omega z \circ V \ dx \ + \ \alpha \int_\Omega z div(V)\ dx \\
& + \int_{\Gamma_{LG}} r \circ V \ ds \ + \ \int_{\Gamma_{LG}} r \ div_\Gamma V \ ds \\
& + \ \mu \int_{\Gamma_{LS}} r \circ V \ dr \ + \ \mu \int_{\Gamma_{LS}} r \ div_\Gamma V \ dr \\
& - \ \delta \int_\Omega \varepsilon \ (r \circ V) \ |\nabla u^\Omega|^2 \ dx \ - \ \delta \int_\Omega \varepsilon \ r \ |\nabla u^\Omega|^2 \ div(V) \ dx \\
& + \ \delta \int_\Omega \varepsilon \ r \ < (\ ^T DV + DV)\nabla u^\Omega, \nabla u^\Omega > \ dx
\end{aligned}$$

with: $div_\Gamma V = (div(V) - < n,\ ^T DV n >)$, n is the external normal and $x = (r, z)$.

ii) the volume constraint $c(\omega)$ *belongs to* $\mathcal{C}^1$ *for all* $\omega = F(\Omega)$, $F \in \mathcal{V}_I$. *And for all* $V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$,

$$\frac{dc}{d\omega}(\Omega).V \;=\; -\int_\Omega r\circ V\; dx \;-\; \int_\Omega r\, div(V)\; dx \tag{16}$$

Proof.
It is done in three steps: 1. Transport of equations; 2. Differentiability with respect to ω; 3. Use of the adjoint technique leading to the expression of the exact differential.

Step 1. Transport of equations. As defined previously, we need to transport the cost function j in order to compute its shape derivative. To this end, we need to transport all the equations on the reference domain $\Omega = F^{-1}(\omega)$.

For any $u,\ v \in X_0(\omega)$, we let:

$$\bar{a}(F;\bar{u},\bar{v}) \;=\; a_{F(\Omega)}(\bar{u}\circ F^{-1}, \bar{v}\circ F^{-1}) \;=\; a_\omega(u,v)$$
$$=\; \int_\Omega \bar{\varepsilon}\;\bar{r} < {}^T(DF^{-1}\circ F)\nabla\bar{u},\ {}^T(DF^{-1}\circ F)\nabla\bar{v} > \ |detDF| d\bar{x}$$

with: $\bar{u} = u\circ F$, $\bar{v} = v\circ F$, $\bar{x} = x\circ F$ and $\bar{\varepsilon} = \varepsilon\circ F$, see Fig. 4.

The mapping $v \in X_0(F(\Omega)) \mapsto v\circ F \in X_0(\Omega)$ is an isomorphism for $F \in \mathcal{F}_0$. In other respect, the Dirichlet's data u_0 is constant, hence $u_0 = u_0 \circ F$. Then, since state equation (5) has an unique solution u^ω, the transported state equation:

$$\text{Find } \bar{u}^F \in X_t(\Omega):\ \bar{a}(F;\bar{u},\bar{v}) = 0,\ \forall \bar{v} \in X_0(\Omega)$$

has an unique solution $\bar{u}^F = u^\omega \circ F$.

Similarly, for any $u \in X_0(\omega)$ we let $\bar{\mathcal{E}}(F;\bar{u}) = \mathcal{E}_{F(\Omega_0)}(\bar{u}\circ F^{-1}) = \mathcal{E}_{\omega_0}(u)$. We have: $\bar{\jmath}(F) = \bar{\mathcal{E}}(F;\bar{u}^F)$,

$$\begin{aligned}\bar{\jmath}(F) \;=\;& \alpha\int_\Omega \bar{z}\ |detDF|\ d\bar{x} \\ &+ \int_{\Gamma_{LG}} \bar{r}\ Jac(F)\ d\bar{s} \;+\; \mu\ \int_{\Gamma_{LS}} \bar{r}\ Jac(F)\ d\bar{r} \\ &-\ \delta\int_\Omega \bar{\varepsilon}\ \bar{r}\ |\ {}^T(DF^{-1}\circ F)\nabla\bar{u}^F|^2\ |detDF|\ d\bar{x}\end{aligned} \tag{17}$$

with $Jac(F) = |detDF|\ \|\ {}^T DF^{-1}.n\|_{I\!R^2}$.

Also, we define:

$$\bar{c}(F) \;=\; \int_B r dx \;-\; \int_\Omega \bar{r}\ |detDF|\ d\bar{x} \;-\; \frac{vol}{2\pi} \tag{18}$$

Step 2. Differentiability with respect to ω.

The mapping $\bar{a}(F;\bar{u},\bar{v})$ is C^1 with respect to $(F;\bar{u})$. It follows from the implicit function theorem that the transported state equation defines a C^1-mapping $F \mapsto \bar{u}^F : \mathcal{F}_0 \to X_t(\Omega)$ in a neighborhood $\mathcal{V}_I$ of I.
Then, since the mapping $\bar{\mathcal{E}}$ is of class $\mathcal{C}^1(\mathcal{F} \times X_0(\Omega))$, the cost function j is continuously differentiable. Also, the constraint function c is continuously differentiable.

Step 3. Expression of the exact differential.

By definition, we have: $\frac{dj}{d\omega}\ (\Omega)\cdot V \ = \ \frac{d\bar{j}}{dF}\ (I)\cdot V\ ,\ \ \forall V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$.
Then, using the classical adjoint technique, we have:

$$\frac{d\bar{j}}{dF}(I).V = \frac{\partial\bar{\mathcal{E}}}{\partial F}(I;u^\Omega).V - \frac{\partial\bar{a}}{\partial F}(I;u^\Omega,p^\Omega).V \quad \forall V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$$

where u^Ω is the solution of the state equation posed in Ω and $p^\Omega \in X_0(\Omega)$ is the adjoint state, unique solution of the following adjoint equation:

$$\frac{\partial\bar{a}}{\partial u}(I;u^\Omega,p^\Omega).v \ = \ \frac{\partial\bar{\mathcal{E}}}{\partial u}(I;u^\Omega).v \quad \forall v \in X_0(\Omega)$$

We have:

$$\frac{\partial\bar{a}}{\partial u}(I;u^\Omega,p^\Omega).v = a_\Omega(p^\Omega,v) \text{ and } \frac{\partial\bar{\mathcal{E}}}{\partial u}(I;u^\Omega).v = -2\delta a_\Omega(u^\Omega,v) = 0 \ \ \forall v \in X_0(\Omega)$$

Hence $p^\Omega \in X_0(\Omega)$ and $a_\Omega(p^\Omega,v) = 0 \ \ \forall v \in X_0(\Omega)$. Therefore: $p^\Omega = 0$.

Hence,

$$\frac{dj}{d\omega}(\Omega).V = \frac{\partial\bar{\mathcal{E}}}{\partial F}(I;u^\Omega).V \quad \forall V \in C^1(\bar{\Omega}, I\!R^2)$$

Using (17) and the classical expression of the derivatives of $|det(DF)|$, $(DF^{-1}\circ F)$ and $(\|\ ^TDF^{-1}.n\|_{I\!R^2})$ -see e.g. ([10], chap. IV)-, we obtain the result i).

The result ii) follows from (18) and the expression of the derivative of $|det(DF)|$. ■

Then, we have straightforwardly

Corollary 1 *At (λ,τ) given in $I\!R \times I\!R^+$, the augmented lagrangian L_τ is locally and continuously differentiable with respect to ω. And for all $V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$,*

$$\frac{\partial L_\tau}{\partial\omega}(\Omega,\lambda).V \ = \ \frac{dj}{d\omega}(\Omega).V \ + \ \lambda\frac{dc}{d\omega}(\Omega).V \ + \ 2\tau\ c(\Omega)\frac{dc}{d\omega}(\Omega).V \tag{19}$$

where $\frac{dj}{d\omega}(\Omega).V$ and $\frac{dc}{d\omega}(\Omega).V$ are defined by (15) and (16) respectively.

5 Discretization

In this section, we discretize the shape derivative of the augmented lagrangian L_τ defined by (19), we define the shape parameters and we obtain the shape gradient. Also, we detail the full optimization process.
Let us recall that the expression $\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V$ depends on u, the unique solution of (5).
Let $(\mathcal{T}_h)$ be a regular family of triangulation, $\omega = \cup_{T \in \mathcal{T}_h} T$, we compute an approximation of u using the classical piecewise linear conforming finite element method (P_1-Lagrange). This finite element approximation is denoted by u_h. The parameter h denotes a characteristic mesh size.

Discretization of the boundary; Shape parameters. Let $\hat{\Omega}$ be an open set of reference; typically $\hat{\Omega}$ is a quarter of a disk, see Fig. 3. The domain of reference $\hat{\Omega}$ is defined using a parametric function:

$$s_{\hat{\Omega}}(t) = \sum_{i=0}^{N-1} \hat{P}_i \, s_i(t) \, , \; t \in [0,1]$$

where $\{s_i(t)\}_{i=0..N-1}$ are piecewise linear functions, $s_i(\frac{j}{N-1}) = \delta_{ij}$; δ_{ij} denotes the Kronecker symbol, and $\hat{P}_i = ((\hat{P}_r)_i, (\hat{P}_z)_i)^T$ are the control points. We set $(\hat{P}_z)_1 = (\hat{P}_z)_0$.

We have $\Omega = \hat{F}_0(\hat{\Omega})$ with $\hat{F}_0 \in \hat{\mathcal{F}}_0$. Similarly, we define the variable boundary Γ_{LG}, the unknown of the problem, by:

$$s_\Omega(t) = \sum_{i=0}^{N-1} P_i \, s_i(t) \, , \; t \in [0,1],$$

Hence, the boundary Γ_{LG} is defined by the N control points $P_i, i = 0..N-1$.
Initially, these points define $\hat{\Gamma}_{LG}$ as follows, see Fig. 5:

$$\hat{P}_i = (0, R)^T$$

$$\hat{P}_i = (\; R \; cos(\frac{(N-1-i)\pi}{2(N-1)}), \; R \; sin(\frac{(N-1-i)\pi}{2(N-1)}) \;)^T \quad i = 2..N-1$$

$$\hat{P}_1 = (R \; cos(\frac{(N-2)\pi}{2(N-1)})/2, R)^T$$

Therefore, during the optimization process, to compute a new domain means to compute new control points $P_i, \; i = 0..N-1$.

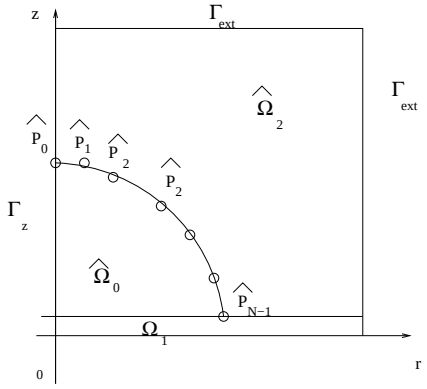

Figure 5: Reference domain. Parametrization.

The shape deformation space. Let us discretize the shape deformation V, $V \in \mathcal{C}^1(\bar{\Omega}, I\!R^2)$. We have: $\Omega = \hat{F}_0(\hat{\Omega})$ with $\hat{F}_0 \in \hat{\mathcal{F}}_0$. We set: $V = \hat{V} \circ \hat{F}_0^{-1}$. V is defined in Ω while $\hat{V}$ is defined in $\hat{\Omega}$.
We approximate $C^1(\bar{\hat{\Omega}}, I\!R^2)$ by $\hat{S}_H$ the vectorial space spanned by $\{\hat{V}_i\}_{i=0..N-1}$:

$$\hat{S}_H \;=\; Span\{\hat{V}_i\}_{i=0..N-1}$$

where the set of vectors $\{\hat{V}_i\}_{i=0..N-1}$ is detailed below.
We set: $H = \frac{1}{N-1}$. The parameter H denotes a characteristic size of the shape deformation space.
Then, the deformation field V is approximated by:

$$V_H = \sum_{i=0}^{N-1} \eta_i V_i \tag{20}$$

where $V_i = \hat{V}_i \circ \hat{F}_0^{-1}$ and $\eta_i,\ i = 0..N-1$, are real coefficients.
We have $V_H = (\hat{V}_H \circ \hat{F}_0^{-1})$ with:

$$\hat{V}_H = \sum_{i=0}^{N-1} \eta_i \hat{V}_i \tag{21}$$

Finally, $C^1(\bar{\Omega}, I\!R^2)$ is approximated by: $S_H = Span\{V_i = \hat{V}_i \circ \hat{F}_0^{-1}\}_{i=0..N-1}$.

The shape deformation basis. We have: $\hat{F}_0 = (I + \hat{V})$ and $\hat{V}$ is approximated by $\hat{V}_H$ defined by (21).

The basis $\{\hat{V}_i\}_{i=0..N-1}$, is defined in $\hat{\Omega}$ as follows. For $i = 0..N-1$, we solve:

$$\left\{\begin{array}{rcll} \Delta(\hat{V}_r)_i & = & 0 & \text{in } \hat{\Omega} \cap Bint \\ (\hat{V}_r)_i & = & 0 & \text{in } \hat{\Omega}/Bint \\ (\hat{V}_r)_i & = & 0 & \text{on } \Gamma_{Gz} \cup \Gamma_{Sz} \\ (\hat{V}_r)_i & = & \frac{(\hat{P}_r)_i}{||\hat{P}_i||}\ s_i & \text{on } \hat{\Gamma}_{LG} \end{array}\right. \tag{22}$$

$$\left\{\begin{array}{rcll} \Delta(\hat{V}_z)_i & = & 0 & \text{in } \hat{\Omega} \cap Bint \\ (\hat{V}_z)_i & = & 0 & \text{in } \hat{\Omega}/Bint \\ (\hat{V}_z)_i & = & 0 & \text{on } \Gamma_0 \cup \hat{\Gamma}_{LS} \cup \hat{\Gamma}_{SG} \\ (\hat{V}_z)_i & = & \frac{(\hat{P}_z)_i}{||\hat{P}_i||}\ s_i & \text{on } \hat{\Gamma}_{LG} \end{array}\right. \tag{23}$$

where $\hat{V}_i = ((\hat{V}_r)_i, (\hat{V}_z)_i)^T$, $\hat{P}_i = ((\hat{P}_r)_i, (\hat{P}_z)_i)^T$ and $\|\hat{P}_i\| = [(\hat{P}_r)_i^2 + (\hat{P}_z)_i^2]^{\frac{1}{2}}$.

Let us notice that in order to extend the deformation vectors all over the domain, we can use an elasticity system instead of a Laplace type equation.

The shape gradient. We approximate V by V_H, see (20), and we have:

$$\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V \approx \frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_H = \sum_{i=0}^{N-1} \eta_i \frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_i$$

Then, the shape gradient denoted by G^H is the vector:

$$G^H = (G_i^H)_{i=0\ldots N-1} = ([\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_i])_{i=0\ldots N-1}$$

$$= ([\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).(\hat{V}_i \circ \hat{F}_0^{-1})])_{i=0\ldots N-1}$$

where $\Omega = \hat{F}_0(\hat{\Omega})$.

Finally, we have, for all $i = 0 \ldots N-1$, see Corollary 1,

$$G_i^H \ = \ \frac{dj}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}) \ + \ \lambda \frac{dc}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}) \ + \ 2\tau\ c(\Omega) \frac{dc}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}) \tag{24}$$

Variables of optimization. Since $\Omega = \hat{F}_0(\hat{\Omega}) = (I + \hat{V})(\hat{\Omega}) \approx (I + \hat{V}_H)(\hat{\Omega})$ with $\hat{V}_H$ defined by (21), and $\hat{V}_i$ defined by (22)-(23), the variables of optimization are the N coefficients $\eta_i,\ i = 0..N-1$.

6 Optimization Process

As mentioned previously, we solve the optimization problem (11) using the Uzawa's algorithm and the Quasi-Newton algorithm BFGS.

- Initially, we set: $\eta_i^0 = 0\,,\ i = 0 \ldots N-1;\ \lambda_0 = 0.$
- We compute $\eta_i^1\,,\ i = 0 \ldots N-1$ such that $j(\eta^1) < j(\eta^0)$ using BFGS.
- We compute the volume constraint $c(\eta^1)$.
- While the volume constraint ($|c(\eta^{k+1})| > eps1$):
  - set $\lambda_{k+1} \ = \ \lambda_k \ + \ \rho\ c(\eta^{k+1})$
  - compute $\eta_i^{k+2}\ \ i = 0..N-1$, such that $j(\eta^{k+2}) < j(\eta^{k+1})$ using BFGS
  - compute the volume constraint $c(\eta^{k+2})$

  Classically, we set $\rho = \tau$, see e.g. [7].

The BFGS algorithm is implemented with constraints of bounds type. The linear search is done using a dichotomic process.
We stop the BFGS algorithm either if $\frac{|j(\eta^{k+2}) - j(\eta^{k+1})|}{j(\eta^{k+1})} < eps2$ or if $\|(G^H)^{k+2}\| < eps3$.

As usual, each call of the algorithm BFGS implies few calls to the simulator.
The simulator does the following:

- it computes the new shape and the new mesh defined by:

$$\Omega = (I + \sum_{i=0}^{N-1} \eta_i \hat{V}_i)(\hat{\Omega})$$

- it solves the state equation (5) posed in Ω by a P_1-Lagrange finite element method (with or without automatic mesh refinement)
- it computes the augmented lagrangian L_τ defined by (9), its gradient G^H defined by (24) and the volume constraint c defined by (10).

The full optimization process is represented in Fig. 6.

7 Structure of the Program ElectroCap

The different modules of the program are indicated in Fig. 7. The arrows (from left to right) means left module use right module. These relations are transitive.

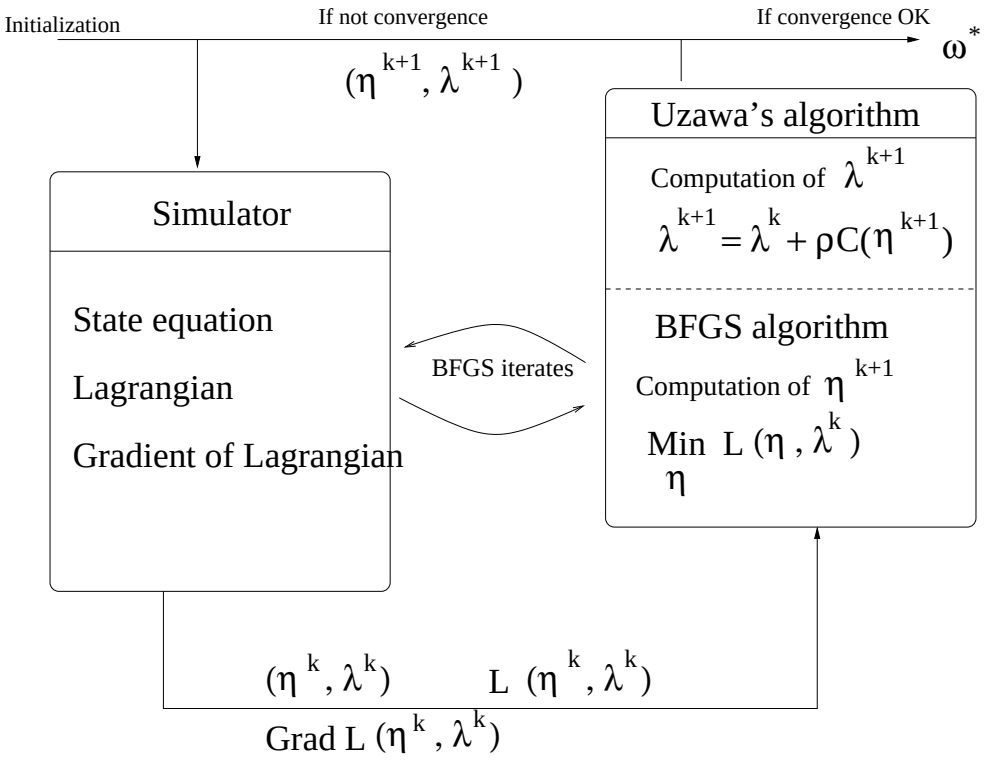

Figure 6: The optimization process

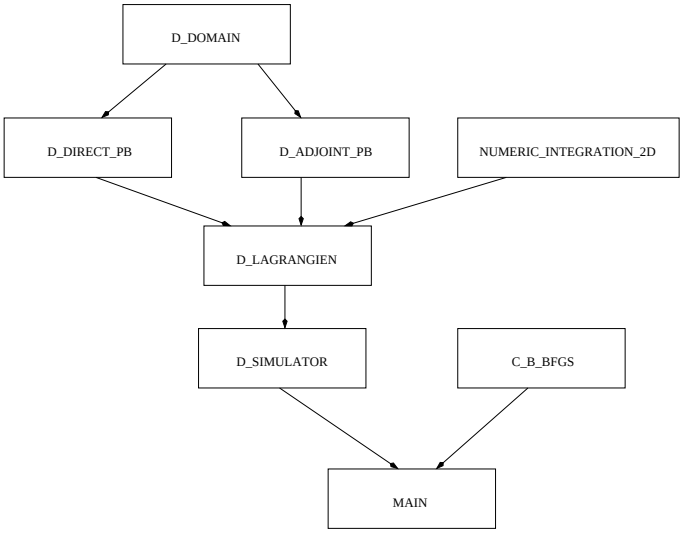

Figure 7: Structure of the Program Electrocap

- Module D_DOMAIN manage all concerning the shape domain we seek to optimize. It describes the geometry, meshes and the transformations V_i , $i = 0 \dots N-1$, associated.
- Module D_DIRECT_PB solve the direct problem.
- Module D_ADJOINT_PB solve the direct problem.
- Module NUMERIC_INTEGRATION_2D implements the integrals (2D and curvilinear) required in the system.
- Module D_LAGRANGIEN implements computation of the augmented lagrangian, the volume constraint and the corresponding shape derivatives.
- Module D_SIMULATOR make the interface with the Uzawa optimizer. It is based on the module D_LAGRANGIEN.
- Module C_B_BFGS is a central module since it implements the Uzawa algorithm. Born constraints are also considered.
- Module MAIN in the main programm.

For more detail on the structure, we refer to the documentation generated by Oxygen, see appendix

8 Numerical Results and Validation

The full optimization process described in the previous section has been implemented in C++. The software *Shapelectrocap* is based on the public C++ finite element library Rheolef [12] and a BFGS algorithm home-developed. The mesh generator used is Bamg [11]. For each simulator call, an automatic mesh refinement is used. This mesh refinement is based on the classical a posteriori estimates, see [11] and references therein. We present in Fig. 8 a typical mesh with the adaptive mesh in the edge.

Numerical data. Numerical data considered are the following:

- The surface tension coefficients(in N/m): $\sigma_{LS} = 2.7\ 10^{-2}$, $\sigma_{LG} = 5\ 10^{-2}$.
- The wetting angle at $u_0 = 0$ (in radians): $\theta_0 = \frac{\pi}{2}$.
- The insulator thickness (in m): $e = 200\ 10^{-6}$,
- The electrical permissivities: $\varepsilon_1 = 2 \times 8.85\ 10^{-12}$ and $\varepsilon_2 = 8.85\ 10^{-12}$.
- The drop volume (in l): $vol = 40\ 10^{-9}$

We assume the Bond number α small i.e. we neglect the gravitational term. We set $\alpha = 0$.

Numerical parameters are the following:

- The penalty parameter: $\tau = \rho = 10^{-3}$.
- The number of control points: $N = 16$.
- The convergence parameter of Uzawa's algorithm: $eps1 = 10^{-3}$.
- The convergence parameter of BFGS algorithm: $eps2 = eps3 = 5\ 10^{-6}$.

Validation of the code. All the code has been validated: direct problem, transport of mesh and shape gradient.
The computed shape gradient have been compared with values obtained by a finite difference method using the following approach. For each shape parameter, a finite difference shape derivative is computed using a domain perturbation of magnitude 10^{-4}. The order of magnitude of the relative error obtained between the two approaches is $10^{-4} - 10^{-6}$, depending on the imposed electrical field value u_0.

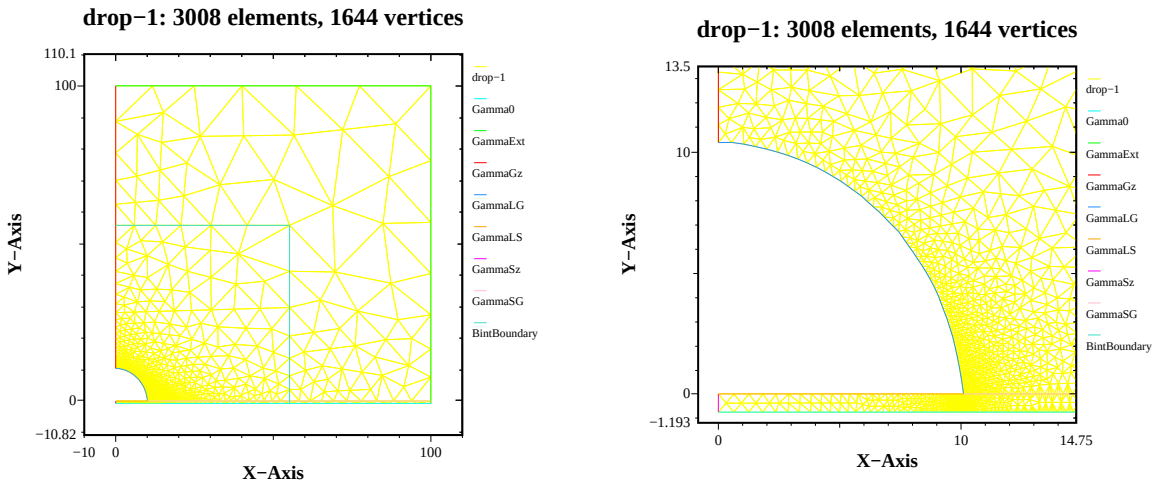

Figure 8: *Left,* Shape and mesh for $u_0 = 400$ V; *Right,* Zoom near the edge.

Drop shape and wetting angle. We present in Fig. 8 the drop shape (with mesh) obtained for $u_0 = 400$ V (left) and a zoom of the refined mesh near the edge (right). All meshes contain approximatively 3000 elements and 1600 vertices.
For each computation, the volume constraint is satisfied at less than 0.1%.

We present the cost function, the augmented lagrangian and its gradient in function of the iterations for $u_0 = 400$ V in Fig. 8. The behavior of the algorithm for the other values of u_0 is similar.

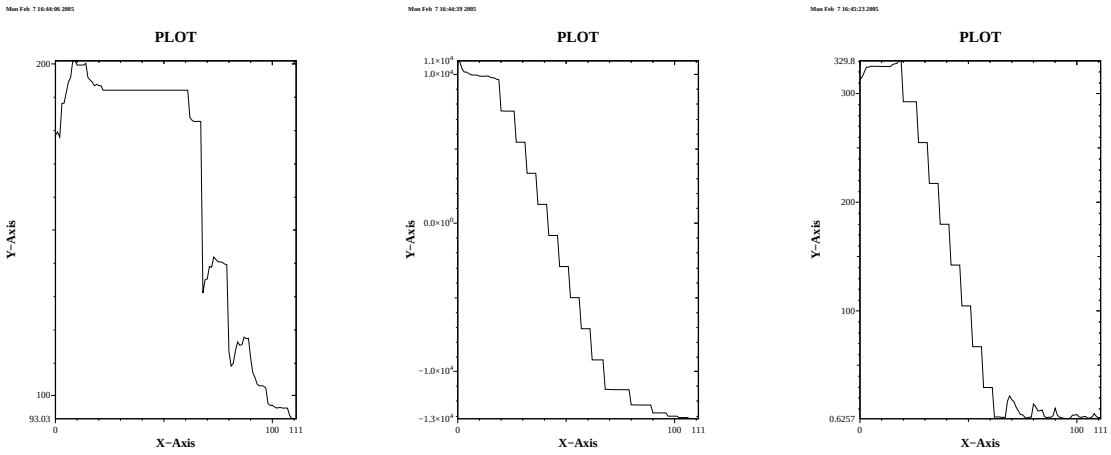

Figure 9: $u_0 = 400$. *Left.* Cost function j vs iterations *Middle.* Augmented Lagrangian L_τ. *Right.* Gradient of L_τ.

We present in Fig. 10, the drop shapes obtained in function of u_0.

We present in Fig. 11, the computed wetting angle values in function of u_0 and the predicted values by the plane capacitor approximation (Lippman's equation).

The good agreement between our software computations and the Lipmann law predictions confirms its validity when the applied electrical potential u_0 is lower than the critical value u_{cr}. For the present physical data, the observed critical value u_{cr} is approximatively 800 V.
For $u_0 \approx 1050$ V the Lippman equation predicts a total spreading of the drop on the substrate: the wetting angle vanishes. Using the present software, and for $u_0 = 1100$ V, we obtain a total spreading of the drop too.

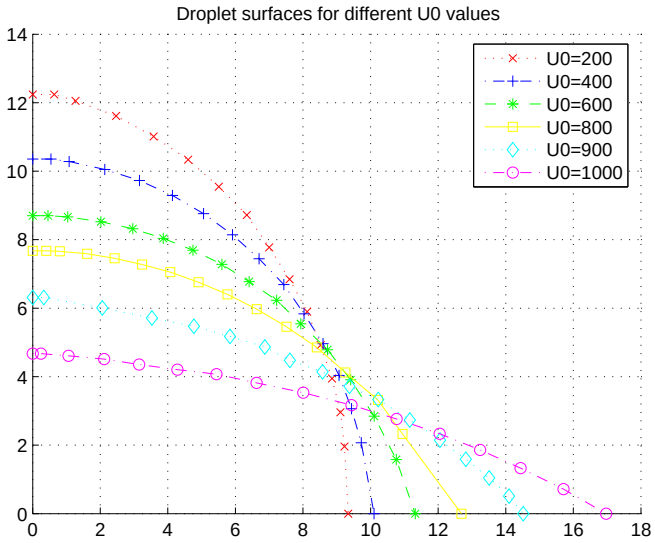

Figure 10: Droplet surfaces for different u_0 values.

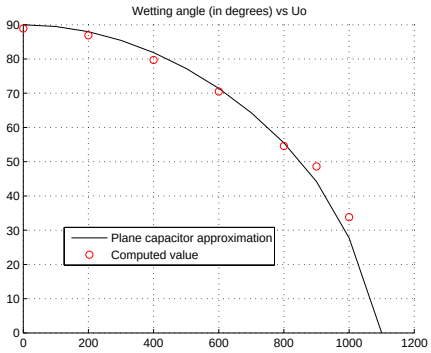

Figure 11: Wetting angle. Computed values and Lippman's equation predictions.

Experimentally, we do not observe a total spreading but a saturation of the wetting angle, see e.g. [13]. The explanation of this locking phenomena is still badly understood by physicists. With the present model, we do not manage to simulate this locking phenomena.

9 Conclusion

We have detailed a global approach modelling an electro-wetting process. The electrified liquid drop is steady-state and its shape is computed by using a shape optimal design method. We compute shapes minimizing the total drop energy. Electrocap, the corresponding software is based on the public mesh generator Bamg and the public C++ finite element library Rheolef.

In a microfluidics point of view, we compared our numerical results with the plane capacitor approximation (Lippman's equation).
For an applied voltage lower than the critical value, one knows that the plane capacitor approximation is valid; and for such values, we retrieved the plane capacitor approximation values.
For an applied voltage greater than the critical value, one knows that the plane capacitor approximation is not valid anymore, it predicts a total spreading instead of a locking phenomena. For such values, we did not observe any locking phenomena with our model. Our model predicted a total spreading too -despite a adaptive mesh refinement near the edge-. This last feature is under investigation.

Acknowledgements. The authors would like to thank C. Quilliet and M. Bienia (Laboratory of Spectrometry Physics -LSP-, Grenoble) for the numerous and fruitful discussions.

A ElectroCap User Manual

We present below the user manual in French since the interactive menu in the code is in french.

A.1 Compilation

Pour compiler le code de la simulation, il faut se placer dans le répertoire principal contenant le fichier **Makefile**, puis lancer la commande **make**. Le code binaire est généré dans le répertoire **bin**, et l'exécutable dans le répertoire **exe**. C'est le fichier **main.exe**.

A.2 Description de l'interface utilisateur

A.2.1 Le menu principal

Lorsqu'on lance l'exécution du programme de simulation, le menu suivant apparait :

```
1 - Editer les parametres du programme
2 - Lancer l'execution du programme
3 - Quitter
```

Si on lance l'exécution du programme sans avoir, auparavant, éditer les paramètres de la simulation, alors le programme utilise des paramètres par défaut.

Lorsqu'on choisit 1, voici le menu qu'on obtient :

```
1 - Editer les parametres du simulateur
2 - Editer les parametres de l'optimiseur
3 - Retour au menu precedent
```

On peut alors, soit modifier les paramètres du simulateur (paramètres physiques du procédé d'électro-mouillage), soit ceux de l'optimiseur, afin d'en améliorer les performances.

A.2.2 Le menu d'édition des paramètres du simulateur

Le menu d'édition des paramètres du simulateur est le suivant :

```
1 - Changer le nombre de points de controle (Ncpt=8)
2 - Changer e l'epaisseur du polymere (e=5.3e-05)
3 - Changer vol le volume de la goutte (vol=6e-08)
```

```
4 - Changer les valeurs des epsilons
    (epsilon1=1.8585e-11  epsilon2=8.85e-12)
5 - Changer la valeur de u0 (u0=0)
6 - Changer les sigma_i
    (sigmaLS=0.02  sigmaLG=0.074  sigmaSG=0.014)
7 - Changer la valeur de la penalisation (tau=0.001)
8 - Changer le nombre d'iterations de la
                    procedure d'adaptation de maillage
                    (nbiteram=2)
9 - Retour au menu precedent
```

On peut, au choix, modifier le nombre de points de contrôle de la forme de la goutte (pris par défaut à 8), changer l'épaisseur du polymère, changer le volume physique de la goutte, changer les permittivités diélectriques dans le polymère ($epsilon_1$) et dans l'air (ϵ_2), changer les valeurs des tensions de surface (σ_{LS}, σ_{LG} et σ_{SG}), changer la valeur de la pénalisation (le terme $\tau C(\omega)^2$ du lagrangien augmenté), ou alors changer la valeur de **nbiteram** qui est le nombre d'itérations de la procédure d'adaptation de maillage lorsqu'on résoud le problème direct du problème d'optimisation de forme.

A.2.3 Le menu d'édition des paramètres de l'optimiseur

Le menu d'édition des paramètres du simulateur est le suivant :

```
1 - Changer la valeur du critere d'arret
                    sur la variable x (epsx=0.1)
2 - Changer le nombre maximum d'iterations
                    sur x (nbmaxx=80)
3 - Changer la valeur du critere d'arret
                    sur lambda (eps=10)
4 - Changer le nombre maximum d'iterations
                    sur lambda (nbmaxl=10)
5 - Changer le nombre maximum global
                    d'iterations (maxiterg=810)
6 - Changer la valeur de lambda_init (lambda_init=0)
7 - Retour au menu precedent
```

La variable **epsx** est le critère d'arrêt de la procédure d'optimisation de la variable **x** (la forme du domaine pour la simulation de l'électro-mouillage) à λ fixe. En effet, la procédure d'optimisation s'arrête lorque, au cours de 5 itérations successives de la procédure, la fonction coût Ct minimisée varie de moins de **epsx**.

La variable **nbmaxx** est le nombre maximum d'itérations qu'on fixe pour l'optimisation de $\mathbf{x}$ à λ fixe.

La variable **eps** est le critère d'arrêt sur la contrainte de volume. Sa valeur correspond à une erreur relative sur le volume de 10^{-2}.

On peut fixer le nombre maximum d'itérations sur λ grâce à la variable **nbmaxl**.

La variable **maxiterg** est le nombre total maximum d'itérations. Elle permet d'arrêter le programme à une itération choisie à l'avance.

La variable **lambda_init** est la valeur initiale de λ.

A.2.4 Le menu de fin de simulation

En fin de simulation, voici le menu proposé :

```
1 - Afficher le lagrangien en fonction
                des iterations
2 - Afficher la norme gradient du lagrangien
                en fonction des iterations
3 - Afficher le cout en fonction des iterations
4 - Afficher la norme du gradient du cout
                en fonction des iterations
5 - Afficher l'evolution de la contrainte de volume
                en fonction des iterations
6 - Afficher l'abscisse du point triple
                en fonction des iterations
7 - Afficher le cosinus de l'angle du point triple
                en fonction des iterations
8 - Afficher u
9 - Afficher le champ E
10 - Afficher la norme de E
11 - Retour au menu precedent
```

Ce menu est assez explicite pour ne pas être commenté.

A.2.5 Résultats de fin de simulation

En fin de simulation, le programme trace la forme de la goutte optimisée, et affiche un certain nombre de résultats. En voici un exemple :

```
Nombre global d'iterations=56
Nombre moyen d'iterations sur x=18.6667
Nombre d'iterations sur lambda=3
Nombre moyen d'iterations de la recherche lineaire=24.9643
Nombre de remaillage=321

lambda=-0.134675

neta[0]=5.97278
neta[1]=5.94897
neta[2]=5.86246
neta[3]=5.74885
neta[4]=5.55465
neta[5]=5.33267
neta[6]=5.04558
neta[7]=4.77101

(theta)t=94.6507 degres
(theta)r=94.6724 degres
cos(theta)t=-0.0810811
cos(theta)r=-0.0814588

temps de calcul : 143 secondes
```

La variable **lambda** contient la valeur de λ finale.

La variable **neta** est la valeur de la variable d'optimisation finale.

La variable **(theta)t** est l'angle de contact théorique (formule approximation condensateur plan) de la goutte sur le polymère.

La variable **(theta)r** est l'angle de contact théorique obtenu par simulation de la goutte sur le polymère.

Enfin, le programme affiche également le temps de calcul réel (pas le temps CPU).

A.3 La documentation OXYGEN

A.3.1 Génération de la documentation

Pour générer la documentation OXYGEN de la simulation, il faut se placer dans le répertoire **prg**. Puis, on lance la commande :

```
doxygen doc/config.txt
```

La documentation se situe dans le répertoire **doc**. Elle est générée en html (documentation complète) et en latex.

A.3.2 Consultation de la documentation html

Pour consulter la documentation html, avec netscape, il faut entrer dans la barre de navigation et charger le fichier
file:/doc/html/index.html

References

[1] B. Berge *Electrocapillarité et mouillage de films isolants par l'eau*, CRAS 317 Série II, (1993) 157.

[2] B. Berge, J. Peseux *Variable focal lens controlled by an external voltage: an application of electrowetting*. Eur. Phys. J. E. **3**, 159-163, (2000).

[3] M. Bienia, C. Quillet, M. Vallade, *Modification of drop shape controlled by electrowetting*, Langmuir **19**, 9328-33, (2003).

[4] B. Bouchereau *Modélisation et simulation numérique de l'electro-mouillage*, PhD Thesis, University of Grenoble I (1997).

[5] Céa J., Conception optimale ou identification de formes - Calcul rapide de la dérivée directionnelle de la fonction coût, *M2AN*, **20**, 3, 371–402 , 1986.

[6] D. Chenais, J. Monnier and J.P. Vila, *A shape optimal design problem with convective and radiative heat transfer. Analysis and implementation.*, J. Optim. Th. Appl., 110,1, (2001).

[7] M. Fortin, R. Glowinski, *Methodes de lagrangien augmente*, Dunod, 1982.

[8] B. Mohammadi, O. Pironneau, *Applied optimal shape design for fluids*, Oxford University Press (2001).

[9] J. Monnier, *Shape sensitivities in a Navier-Stokes flow with convective and gray bodies radiative thermal transfer*, Optim. Control. Appl. Meth., 24, pp 237-256, (2003).

[10] F. Murat and J. Simon, *Sur le Contrôle par un Domaine Géométrique*, Publication of the Laboratory of Numerical Analysis, University Paris VI, 1976.

[11] F. Hecht, *Freeware Bamg (Bidimensional Anisotrope Mesh Generator)*, http://www-rocq1.inria.fr/gamma, INRIA.

[12] P. Saramito, N. Roquet and J. Etienne, *Freeware Rheolef*, http://www-lmc.imag.fr/lmc-edp/Pierre.Saramito/rheolef/, LMC-IMAG.

[13] M. Vallet, M. Vallade and B. Berge, *Limiting phenomena for the spreading of water on polymer films by electrowetting*, Eur. Phys. J. B, **11**, 583, (1999).

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399